**SCHOOL MANAGEMENT SYSTEM**

**PROBLEM STATEMENT:**

To develop software model on “SCHOOL MANAGEMENT SYSTEM” using software rose with various UML diagram.

**PROBLEM DESCRIPTION:**

The main description of SMS is to create a computerized system for effectively and efficient managed the academic performance of the student. The priority of the system is easy to use, simply yet able to keep information organized. Hopefully the system will act as the first step towards realizing the dream of becoming a paperless society. School administrator or teacher use manual technique widely to compute examination mark and to produce grade, to calculate total mark and also to view statistic for student information. There is also need to upgrade the system using the computer based information system

**3) SRS:**

**a)INTROUCTION**

1. **PURPOSE**

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to users. In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1. **SCOPE**

Primarily, the scope pertains to the medical expert system. It focuses on patients and helps them to get relieved from the disease by providing remedies. The system is used all hospitals. This system is used because it is user friendly. The system gives the treatment of respected disease. This system can be used with the minimum requirement specified. So this does not occupy more space.

1. **DEFINITION,ACRONYMS AND ABBREVIATIONS**

SMS- School Management System

SRS- Software Requirement Specification

GUI- Graphical User Interface

NetBeansIDE-8.1

Microsoft Office Access 2007 Database

1. **SYSTEM REQUIREMENT SPECIFICATION**

**HARDWARE**:

PROCESSOR: PENTIUM IV 2.6 GHZ

RAM: 512 MB DD RAM

MONITOR: 15’’ COLOR

HARD DISK: 250GB

CDDRIVE: LG52X

KEYBOARD: SATNDARD 102 KEYS

MOUSE: OPTICAL MOUSE

**SOFTWARE:**

FRONT END: NetBeans IDE

BACK END: New Microsoft Office Access 2007

OPERATING SYSYEM: WINDOWS XP, WINDOWS 8, WINDOWS 10,

**b)GENERAL DESCRIPTION**

The proposed system that is being developed is user friendly system. The processing speed is very high when compared to the existing system. The space occupied by the proposed system in the memory is also very less. School Management Software is a complete and customized solution for a school handling regular management work. By going through the flow of application one can easily maintain its student detail, fees detail, Employee detail, Certificate details, Time Table and Exam scheduling, Exams and Question Paper detail and Transportation Detail. This software includes the management of students, Employee, Accounts, Transportation, library, Fees, Exam, Time Table, Inventory and its respective reports with administrative module to provide better security. User-friendly menus can easily generate all the reports.

**c)SPECIFIC REQUIREMENTS**

1. **EXTERNAL INTERFACE REQUIREMENTS:**

* **USER INTERFACE REQUIREMENTS:**

The user interface (UI) is everything designed into an information device with which a human being may interact -- including display screen, keyboard, mouse, light pen, the appearance of a desktop, illuminated characters, help messages, and how an application program or a Web site invites interaction and responds to it. In early computers, there was very little user interface except for a few buttons at an operator's console. The user interface was largely in the form of punched card input and report output. The user interface can arguably include the total "user experience," which may include the aesthetic appearance of the device, response time, and the content that is presented to the user within the context of the user interface.

1. Login Module - The login module helps to get into the staff system. The user name and password is obtained from the user and is verified and then goes on to the next screen.
2. Selection module -The selection module is used to select one of the available options. The available options are student and staff details. Each module is clicked the respective module opens.
3. Staff Detail module - In this module staff enter details about the students academics performance and submit the record to receive remedy from system.
4. Student Login module - This Login module only for students to give rnn and submit. Then it displays the student academics details which they posted and recorded in database.

* **HARDWARE INTERFACE REQUIREMENTS**

|  |  |
| --- | --- |
| Ram | 1GB |
| Hard Disk | 160GB |
| Processor | Pentium-4 |
| Operating System | Win XP/2000/98 |
| Mother Board | Intel d856 |
| Keyboard | Microsoft keyboard |
| Mouse | Microsoft optical |
|  |  |
| Monitor | LG 17”flatiron CRT |
|  |  |
|  |  |

* **SOFTWARE INTERFACE REQUIREMENTS**

OPERATING SYSTEM: WIN2000/XP

PROGRAMMING PACKAGE: Netbeans IDE-Java

DATABASE: New Microsoft Office Access 2007

DESIGING SOFTWARE: RATIONAL ROSE 4.2.0

### 

### **COMMUNICATION INTERFACE**

The MES shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

1. **FUNCTIONAL REQUIREMENTS:**

**USER LOGIN:**

The login module helps to get into the school management system. The user id and password is obtained from the user and is verified and then goes on to the next screen.

**SELECTION MODULE:**

The selection module is used to select one of the available options. The available options are student and staff details. Each module is clicked the respective module opens.

**Staff Detail module:**

In this module staff enters details about the students academics and submit the record to receive remedy from system.

**Student Login module**

This Login module only for students to give rnn and submit. Then it displays the student academics details which they posted and recorded in database

**USE CASES**

**Use Case1: User Login**

The login module helps to get into the School management system. The user name and password is obtained from the user and is verified and then goes on to the next screen.

**ACTOR:**

Staff.

**INPUT:**

The Staff beings enter user name and password.

**OUTPUT:**

The user can access the system and directed to selection module.

**Use Case2: Selection**

The selection module is used to select one of the available options. The available options are staff and student details. Each module is clicked the respective module opens.

**ACTOR:**

Student and staff.

**INPUT:**

The user can select the respective module.

**OUTPUT:**

The users are directed to selected modules.

**Use Case3 Staff Detail module:**

In this module staff enters details about the students academics and submit the record to receive remedy from system.

**ACTOR:**

Staff

**INPUT:**

The staff gives the student name, marks etc.

**OUTPUT:**

The data is stored in the database of the system to get remedies.

**Use Case4: Student’s Details**

In this module students can view his/her from staff through system.

**ACTOR:**

Student

**INPUT:**

The data is retrieved from the database.

**OUTPUT:**

The marks and attendance for the rrn is given to user.

1. **NON FUNCTIONAL REQUIREMENTS:**

* **Security**

The system use SSL (secured socket layer) in all transactions that include any confidential student information. The system must automatically log out all students after a period of inactivity. The system should not leave any cookies on the staff’s computer containing the user’s password. The system’s back-end servers shall only be accessible to authenticated administrator. Sensitive data will be encrypted before being sent over insecure connections like the internet.

* **Reliability**

The system provides storage of all databases on redundant computers with automatic switchover. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

* **Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

* **Maintainability**

A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

* **Portability**

The application is NetBaeans IDE and scripting language based. So the end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user can use this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops, and PDA etc.

1. **DESIGN CONSTRAINTS**

The system shall be built using a standard web page development tool that conforms to Microsoft’s GUI standards like NetBeans, XML etc.

* **Hardware Limitations:**

There must be 64 MB on board memory

* **Control Functions:**

The software must be very user friendly and display appropriate error messages.

* **Interfaces to other applicants**:

Not applicable

* **Parallel operations**:

It must support many users simultaneously.

* **Reliability requirements:**

Data redundancy and use of special/blank characters must be avoided.

* **Safety/security considerations**:

The applicant must be exited always normally.

1. **LOGICAL DATABASE REQUIREMENTS**

Database is a place/container where all the data is stored. The database management system used in the system is Microsoft Office Access 2007 Data baseis. Microsoft Office Access 2007 Database an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system)(RDBMS).Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)' daughter, and ‘Microsoft Office Access 2007 Database", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). In a database, we would be grouping only related data together and storing them under one group name called table. This helps in identifying which data is stored and under what name. It reduces the time to search for a particular data in a whole database. The SMS must be able to use several data formats according to the data formats that are provided by the data bases of expert systems. A transaction should have all the properties of a data base transaction (Atomicity, Consistency, Isolation, Durability).

1. **OTHER REQUIREMENTS**

NIL

**4) DESIGN AND IMPLEMENTATION:**

**UML:**

The **Unified Modeling Language** is a standard visual modeling language intended to be used for

* modeling business and similar processes,
* analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML can be applied to diverse **application domains** (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component **software development methods** and for various **implementation platforms** (e.g., J2EE, .NET).

UML is a standard modeling **language**, not a **software development process.**

* provides guidance as to the order of a team’s activities,
* specifies what artifacts should be developed,
* directs the tasks of individual developers and the team as a whole, and
* offers criteria for monitoring and measuring a project’s products and activities.

**TYPES OF UML DIAGRAMS:**

* Use-case diagrams
* Sequence diagrams
* Collaboration diagrams
* Activity diagrams
* Class diagrams
* State diagrams

**USE CASE DIAGRAM**

**DEFINITON:**

A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" to essential processes, as well as the relationships among different use cases. Use-case diagrams generally show groups of use cases -- either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality. A use-case diagram is typically used to 6communicate the high-level functions of the system and the system's scope.

**DESCRIPTION:**

Actor

Use case

Include

Generalization

Extends

**USE CASE DIAGRAM FOR SCHOOL MANAGEMENT SOFTWARE:**

:



Fig 1

**EXPLANATION**:

A use case diagram is a graphic depiction of the interactions among the elements of a system.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term “system” refers to something being developed or operated, such as a mail-order product sales and service web site. Use case diagrams are employed in UML (Unified Modeling Language, a standard notation for the modeling of real-world objects and systems.

Use-case diagram is used to capture the dynamic behavior of the model. The internal and external agents here are known as actors. So, a use case diagram consists of actors, use cases and their relationships. In the above use case diagram, there are three actors namely “administrator, staff, student”. They are related each other by various use cases such as student views course and course calendar. Those activities are carried under the control of administrator. So; this describes the relationship between staff, student and administrator in various use cases.

**SEQUENCE DIAGRAM**

**DEFINITION:**

Sequence diagrams show a detailed flow for a specific use case. They show the calls between the different objects in their sequence. A sequence diagram has two dimensions: the **vertical dimension** shows the sequence of messages/calls in the time order that they occur, the **horizontal dimension** shows the object instances to which the messages are sent. A sequence diagram is very simple to draw. Across the top of your diagram, identify the class instances. If a class instance sends a message to another class instance, draw a line with an open arrowhead pointing to the receiving class instance; place the name of the message/method above the line. Optionally, for important messages, you can draw a dotted line with an arrowhead pointing back to the originating class instance label the return value above the dotted line.

**DESCRIPTION**

Actor

Class instance

Connection lines

Messages

**SEQUENCE DIAGRAM FOR SCHOOL MANAGEMENT SOFTWARE:**



Fig 2

**EXPLANATION:**

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects need to carry out the functionality of the scenario. Sequence diagram are typically associated with use case realizations in the Logical View of the system under development. Sequence diagram are sometimes called event diagrams or event scenarios.

Sequence diagram show the interaction between participating objects in a given use case. They are helpful to identify the missing objects that are not identified in the analysis object model. To see the interaction between objects, the following describe the sequence diagram of each identified use cases.

Admin is the control objects that manage the activities of the student, teacher, class and subject. The class and subject will be having the self connections. The attendance and report cards will be maintained by the teacher. Each object will be having a return function to the Admin.

**COLLOBORATION DIAGRAM**

**DEFINITION:**

Collaboration diagrams provide a view of the interactions or structural relationships between objects in the current model. The collaboration diagram emphasizes the relationship between objects whereas sequence diagrams emphasize the sequence of events. Collaboration diagrams contain objects, links, and messages. Use collaboration diagrams as the primary vehicle to describe interactions that express decisions about system behavior.

**DESCRIPTION**

Actor

Class instance

Connection lines

Messages

**COLLOBORATION DIAGRAM FOR SCHOOL MANAGEMENT SYSTEM**



Fig 3

**EXPLANATION:**

Like sequence diagrams, collaborations diagram are also interactions diagrams. Collaborations diagrams convey the same information as sequence diagrams, but focus on object roles instead of the times that messages are sent. In a sequence diagrams, objects roles are the vertices and messages are the connecting links.

A collaborations diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

In this collaborations diagram, the user manages the course. The course administrator has some of the following activities like: they will create course and topic. Finally; the course administrator will assign a staff to the respective course.

**ACTIVITY DIAGRAM**

**DEFINITION**

Activity diagrams show the procedural flow of control between two or more class objects while processing an activity. Activity diagrams can be used to model higher-level business process at the business unit level, or to model low-level internal class actions. Activity diagrams are best used to model higher-level processes. This is because activity diagrams are "less technical" in appearance, compared to sequence diagrams. The activity is modeled by drawing a rectangle with rounded edges, enclosing the activity's name. Activities can be connected to other activities through transition lines, or to decision points that connect to different activities guarded by conditions of the decision point. Activities that terminate the modeled process are connected to a termination point. Optionally, the activities can be grouped into swim lanes.  THERE CAN BE ONLY ONE START AND ONE STOP.

**DESCRIPTION**

Start

Stop

Swim lanes

Rectangular rounded edged box

Decision box-diamond

Merge

Fork

Join

Connection lines

**ACTIVITY DIAGRAM FOR SCHOOL MANAGEMENT SYSTEM**



Fig 4

**EXPLANATION**:

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language (UML), activity diagram are intended to model both computational and organizational processes (i.e. Workflows). Activity diagrams show the overall flow of control.

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagram are prepared to capture the entire flow in a system. There are initial state and final state in the activity diagram which denotes the stages of the task.ie. Creation of course and completion of course. There are also decision branches which has guards. In this diagram, the decision branch is course and the guards are “new course and course exists”. A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.

A join node joins multiple concurrent flows back into single outgoing flow. A fork and join mode used together are often referred to as synchronization. Here there is synchronization between various control flows.

**CLASS DIAGRAM**

**DEFINITION:**

The class diagram shows how the different entities relate to each other. It shows the static structures of the system. A class diagram can be used to display logical classes, which are typically the kinds of THINGS, the business people in an organization talk about. Class diagrams can also be used to show implementation classes, which are the things that programmers typically deal with. An implementation class diagram will probably show some of the same classes as the logical class diagram. The implementation class diagram won't be drawn with the same attributes, however, because it will most likely have references to things like Vectors and Hash Maps.

**DESCRIPTION:**

Rectangular box with three horizontal sections

Generalization

Association

Aggregation

Navigation

Multiplicity

**CLASS DIAGRAM FOR SCHOOL MANAGEMENT SYSTEM**



Fig 5

**EXPLANATION:**

A class diagram in the Unified Modeling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is a static diagram. It represents the static view of an application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagram is a collection of classes, interfaces, associations, collaborations and constraints. In the above class diagram, there are six classes such as Course admin, Student, Staff, Course, Course Calendar and Topic. Under each class the corresponding attributes and functions are mentioned. For example, under the “class name course”, the attributes are “course name, coursed”, the functions are “viewing, modifying, removing and creating courses”.

**STATE DIAGRAM:**

State diagram describes different states of a component in a system. The states are specific to a component/object of a system. A State chart diagram describes a state machine.

**PURPOSE :**

State chart diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. State chart diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model life time of an object from creation to termination.

State chart diagrams are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purposes of using State chart diagrams:

* To model dynamic aspect of a system.
* To model life time of a reactive system.
* To describe different states of an object during its life time.
* Define a state machine to model states of an object.

**5 ) CODING:**

**FRONT END (Netbeans IDE& java )**

**1.LOGIN SCREEN**

Home page

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class home extends javax.swing.JFrame {

/\*\*

\* Creates new form home

\*/

public home() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton2 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(null);

jPanel1.setBackground(new java.awt.Color(204, 255, 255));

jPanel1.setPreferredSize(new java.awt.Dimension(1920, 2000));

jPanel1.setLayout(null);

jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton2.setText("Staff");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPanel1.add(jButton2);

jButton2.setBounds(250, 120, 120, 80);

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Student");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.add(jButton1);

jButton1.setBounds(50, 120, 110, 80);

jLabel1.setBackground(new java.awt.Color(255, 0, 51));

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

jLabel1.setForeground(new java.awt.Color(0, 102, 102));

jLabel1.setText("SAK Higher Secondary School");

jPanel1.add(jLabel1);

jLabel1.setBounds(60, 30, 340, 60);

getContentPane().add(jPanel1);

jPanel1.setBounds(0, 0, 450, 300);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new listStudent(). setVisible (true); // TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new staffLogin().setVisible (true); // TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(home.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new home().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

// End of variables declaration

private void move() {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}

Login page

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

/\*\*

\* Creates new form Login

\*/

Statement st;

public Login() {

initComponents();

connect();

}

public void Clear()

{

jPasswordField1.setText("");

jTextField1.setText("");

}

void connect()

{

try{

Connection con=DriverManager.getConnection("jdbc:ucanaccess://E:/Desktop/School Managment new/schoolmgmt.accdb");

st=con.createStatement();

JOptionPane.showMessageDialog(null," Database Connected");

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jTextField1 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jPasswordField1 = new javax.swing.JPasswordField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(null);

jPanel1.setBackground(new java.awt.Color(204, 255, 255));

jPanel1.setLayout(null);

jButton1.setBackground(new java.awt.Color(204, 255, 255));

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Sign In");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.add(jButton1);

jButton1.setBounds(160, 230, 130, 25);

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jPanel1.add(jTextField1);

jTextField1.setBounds(220, 100, 190, 40);

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel3.setText("Password");

jPanel1.add(jLabel3);

jLabel3.setBounds(60, 160, 90, 20);

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel2.setText("Login In");

jPanel1.add(jLabel2);

jLabel2.setBounds(60, 100, 100, 40);

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N

jLabel1.setText(" SAK Higher Secondary School");

jPanel1.add(jLabel1);

jLabel1.setBounds(70, 20, 350, 60);

jPanel1.add(jPasswordField1);

jPasswordField1.setBounds(220, 150, 190, 40);

getContentPane().add(jPanel1);

jPanel1.setBounds(0, 0, 500, 310);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

// String user1=(String)jTextField1.getText();

String username = jTextField1.getText().trim();

String password = jPasswordField1.getText().trim();

String sql= "select username,password from login where username='"+username+"'and password='"+password+"'";

ResultSet rs=st.executeQuery(sql);

int count=0;

while(rs.next())

{

count=count +1;

}

if (count==1)

{

JOptionPane.showMessageDialog(null,"User Found, Access Granted!");

close();

staffLogin b=new staffLogin();

b.setVisible(true);

}

else if (count>1)

{

JOptionPane.showMessageDialog(null,"Duplicate User, Access Denied!");

}

else

{

JOptionPane.showMessageDialog(null,"User Not Found");

Clear();

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

} // TODO add your handling code here:

// TODO add your handling code here:

// TODO add your handling code here:

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public void close()

{

WindowEvent winClosingEvent=new WindowEvent(this,WindowEvent.WINDOW\_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

Staff login

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Ammarah

\*/

public class staffLogin extends javax.swing.JFrame {

/\*\*

\* Creates new form staffLogin

\*/

public staffLogin() {

initComponents();

//jLabel4.setVisible(false);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPasswordField1 = new javax.swing.JPasswordField();

jTextField1 = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

getContentPane().setLayout(null);

jPanel1.setBackground(new java.awt.Color(204, 255, 255));

jPanel1.setLayout(null);

jPanel1.add(jPasswordField1);

jPasswordField1.setBounds(190, 160, 160, 30);

jPanel1.add(jTextField1);

jTextField1.setBounds(190, 100, 160, 30);

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel2.setText("Password");

jPanel1.add(jLabel2);

jLabel2.setBounds(70, 160, 70, 40);

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N

jLabel3.setText("Staff Login");

jPanel1.add(jLabel3);

jLabel3.setBounds(150, 40, 120, 20);

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Signin");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel1.add(jButton1);

jButton1.setBounds(150, 220, 80, 30);

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel1.setText("Login");

jPanel1.add(jLabel1);

jLabel1.setBounds(70, 100, 70, 40);

jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel4.setForeground(new java.awt.Color(255, 0, 51));

jLabel4.setText("Login sucessfully ");

jPanel1.add(jLabel4);

jLabel4.setBounds(120, 70, 120, 17);

getContentPane().add(jPanel1);

jPanel1.setBounds(0, 0, 450, 290);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//jLabel4.setVisible(true);

EnterStudentDetails esd = new EnterStudentDetails();

esd.setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(staffLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(staffLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(staffLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(staffLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new staffLogin().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JTextField jTextField1;

// End of variables declaration

}

Student loging page

import java.awt.HeadlessException;

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class listStudent extends javax.swing.JFrame {

/\*\*

\* Creates new form listStudent

\*/

Statement st;

Connection conn;

PreparedStatement pst;

ResultSet rs;

String rrn;

public listStudent() {

initComponents();

//jLabel9.setVisible(false);

connect();

}

public void connect()

{//Connection con=null;

try{

conn=DriverManager.getConnection("jdbc:ucanaccess://E:/Desktop/School Managment new/schoolmgmt.accdb");

st=conn.createStatement();

//JOptionPane.showMessageDialog(null,"Connected");

}

catch(SQLException | HeadlessException e){

JOptionPane.showMessageDialog(null,e);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setBackground(new java.awt.Color(204, 255, 255));

getContentPane().setLayout(null);

jPanel2.setBackground(new java.awt.Color(204, 255, 255));

jPanel2.setLayout(null);

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N

jLabel2.setText("Academics");

jPanel2.add(jLabel2);

jLabel2.setBounds(162, 11, 146, 42);

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jPanel2.add(jTextField1);

jTextField1.setBounds(120, 60, 190, 20);

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel3.setText("RRN");

jPanel2.add(jLabel3);

jLabel3.setBounds(50, 60, 50, 17);

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Submit");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel2.add(jButton1);

jButton1.setBounds(340, 60, 77, 25);

jPanel1.setBackground(new java.awt.Color(153, 153, 255));

jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel4.setText("Name");

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel1.setText("Mark");

jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel5.setText("Attendance");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jTextField3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField3ActionPerformed(evt);

}

});

jTextField4.setText("jTextField4");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel4)

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(66, 66, 66)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 162, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT\_SIZE, 180, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jTextField4, javax.swing.GroupLayout.DEFAULT\_SIZE, 161, Short.MAX\_VALUE)

.addComponent(jTextField3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel4)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 26, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 28, Short.MAX\_VALUE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(18, Short.MAX\_VALUE))

);

jPanel2.add(jPanel1);

jPanel1.setBounds(30, 110, 380, 160);

jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel9.setForeground(new java.awt.Color(204, 0, 51));

jLabel9.setText(" Executed sucessfully ");

jPanel2.add(jLabel9);

jLabel9.setBounds(120, 90, 190, 17);

jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton2.setText("Back");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPanel2.add(jButton2);

jButton2.setBounds(340, 90, 73, 23);

getContentPane().add(jPanel2);

jPanel2.setBounds(0, 0, 460, 290);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try{

st=conn.createStatement();

String rrn=(String)jTextField1.getText();

String s = "select studentname,score,attendence FROM staff where rrn='"+rrn+"' ";

rs=st.executeQuery(s);

//JOptionPane.showMessageDialog(null, "entered");

while(rs.next())

{

//JOptionPane.showMessageDialog(null, "entered");

jTextField3.setText(rs.getString("studentname"));

jTextField4.setText(rs.getString("score"));

jTextField2.setText(rs.getString("attendence"));

}

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new home().setVisible (true);

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new listStudent().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

// End of variables declaration

}

List of student detail

import java.awt.HeadlessException;

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

import java.sql.DriverManager;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class listStudent extends javax.swing.JFrame {

/\*\*

\* Creates new form listStudent

\*/

Statement st;

Connection conn;

PreparedStatement pst;

ResultSet rs;

String rrn;

public listStudent() {

initComponents();

//jLabel9.setVisible(false);

connect();

}

public void connect()

{//Connection con=null;

try{

conn=DriverManager.getConnection("jdbc:ucanaccess://E:/Desktop/School Managment new/schoolmgmt.accdb");

st=conn.createStatement();

//JOptionPane.showMessageDialog(null,"Connected");

}

catch(SQLException | HeadlessException e){

JOptionPane.showMessageDialog(null,e);

}

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel2 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jPanel1 = new javax.swing.JPanel();

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setBackground(new java.awt.Color(204, 255, 255));

getContentPane().setLayout(null);

jPanel2.setBackground(new java.awt.Color(204, 255, 255));

jPanel2.setLayout(null);

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N

jLabel2.setText("Academics");

jPanel2.add(jLabel2);

jLabel2.setBounds(162, 11, 146, 42);

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jPanel2.add(jTextField1);

jTextField1.setBounds(120, 60, 190, 20);

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel3.setText("RRN");

jPanel2.add(jLabel3);

jLabel3.setBounds(50, 60, 50, 17);

jButton1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton1.setText("Submit");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jPanel2.add(jButton1);

jButton1.setBounds(340, 60, 77, 25);

jPanel1.setBackground(new java.awt.Color(153, 153, 255));

jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel4.setText("Name");

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel1.setText("Mark");

jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel5.setText("Attendance");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jTextField3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField3ActionPerformed(evt);

}

});

jTextField4.setText("jTextField4");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel4)

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(66, 66, 66)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 162, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT\_SIZE, 180, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jTextField4, javax.swing.GroupLayout.DEFAULT\_SIZE, 161, Short.MAX\_VALUE)

.addComponent(jTextField3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel4)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 26, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 28, Short.MAX\_VALUE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(18, Short.MAX\_VALUE))

);

jPanel2.add(jPanel1);

jPanel1.setBounds(30, 110, 380, 160);

jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jLabel9.setForeground(new java.awt.Color(204, 0, 51));

jLabel9.setText(" Executed sucessfully ");

jPanel2.add(jLabel9);

jLabel9.setBounds(120, 90, 190, 17);

jButton2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

jButton2.setText("Back");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPanel2.add(jButton2);

jButton2.setBounds(340, 90, 73, 23);

getContentPane().add(jPanel2);

jPanel2.setBounds(0, 0, 460, 290);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

try{

st=conn.createStatement();

String rrn=(String)jTextField1.getText();

String s = "select studentname,score,attendence FROM staff where rrn='"+rrn+"' ";

rs=st.executeQuery(s);

//JOptionPane.showMessageDialog(null, "entered");

while(rs.next())

{

//JOptionPane.showMessageDialog(null, "entered");

jTextField3.setText(rs.getString("studentname"));

jTextField4.setText(rs.getString("score"));

jTextField2.setText(rs.getString("attendence"));

}

}

catch(Exception ex){

JOptionPane.showMessageDialog(null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new home().setVisible (true);

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(listStudent.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new listStudent().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

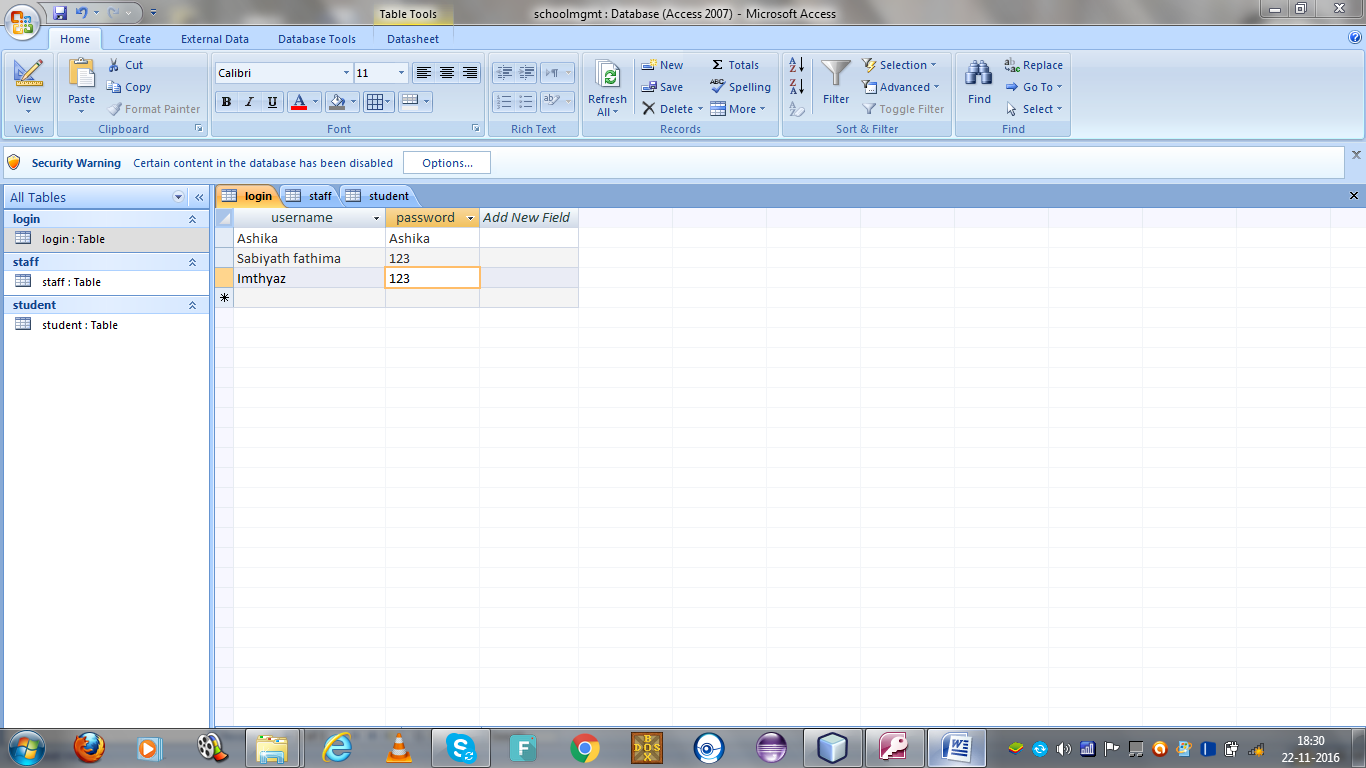
// End of variables declaration

}

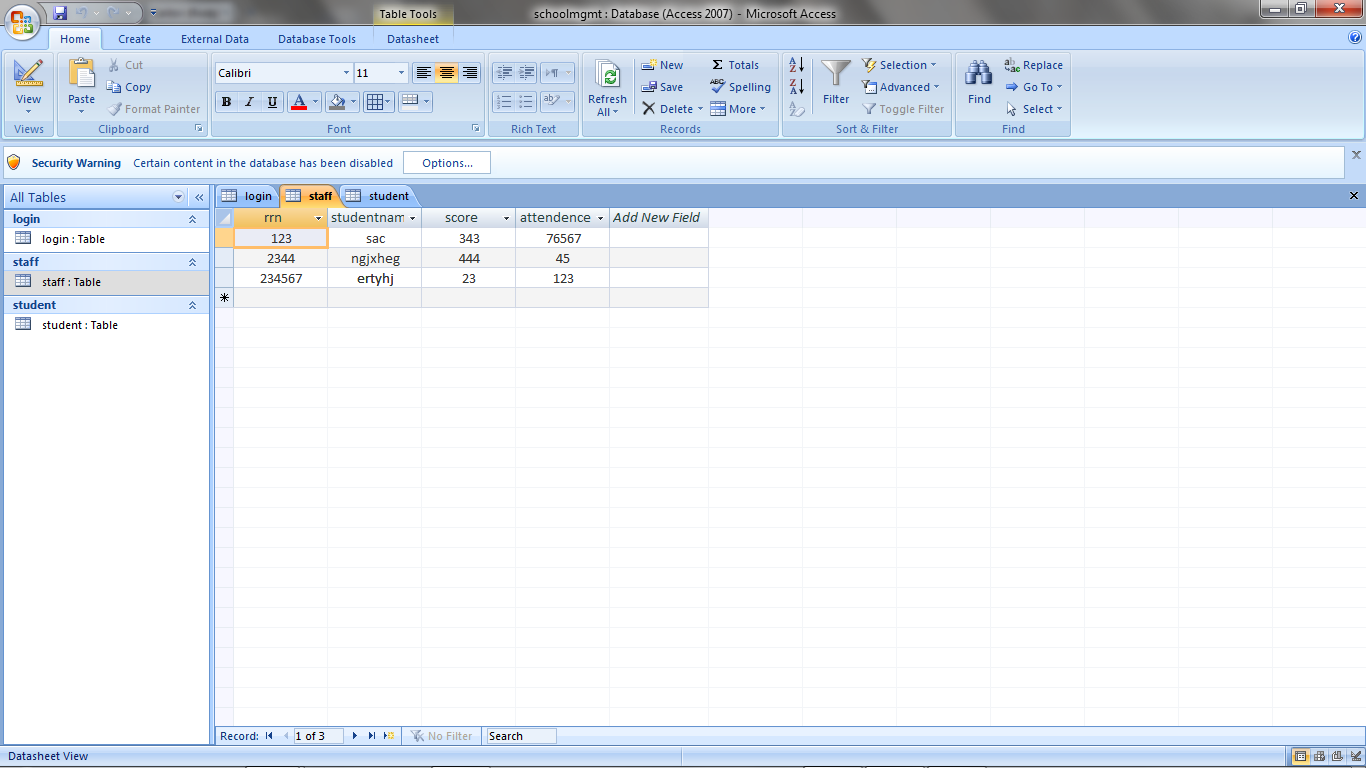
**DATABASE:**

**TABLES:**

LOGIN DETAILS:



STAFF



RISK MANAGEMENT

The privacy and production are risk factors of our project. If the student tries to snoop inside the system to change their marks or attendance, it is a highly riskable issue which will be faced by the administration. To overcome this risk factor, the administration can use firewall to avoid accessibility of others apart from the administration.

CONCLUSION

The School Management System is developed using software rational rose using UML diagram.